Traitement de formulaires en PHP

Quelques rappels

- Nous avons vu dans ce chapitre d'introduction le mécanisme fondamental de création automatique d'une variable PHP pour chaque information en provenance d'un formulaire lors de sa validation, facilité qui assure à PHP son succès.
- L'interpréteur PHP appelé par la validation d'un formulaire HTML a accès immédiat aux données des différents champs de ce formulaire.
  La règle est simple : le nom donné à un élément de formulaire grâce au paramètre NAME devient alors une variable PHP portant le même nom, mais précédée du symbole $ (ce symbole distingue toujours les identificateurs de variables)
- Ainsi, chaque NAME="abc" définit automatiquement une variable $abc. Sa valeur PHP est égale au contenu du champs HTML, identifié par "NAME=abc"
- Par exemple, soit le simple formulaire :

```
<FORM ACTION="inscrire.php" >
<INPUT TYPE="text" NAME="prenom" SIZE=10 VALUE="Jean">
<INPUT TYPE="submit"  VALUE="Valider">
<FORM >
```

  La validation provoque le transfert des données du formulaire (nature, nom et valeur des éléments) vers le programme appelé *inscrire.php* .
  Celui-ci possède d'emblée une variable $prenom dont la valeur est la chaine saisie à la validation, ou "Jean" par défaut.

Interactivité fondamentale entre HTML et PHP

On distingue 2 façons de passer des valeurs issues d'un formulaire à un script écrit en PHP (ou un autre langage). Il s'agit de méthodes implémentées dans le protocole HTTP appelées GET et POST

1. **méthode GET**

   - Il s'agit du mode de transmission par défaut . Les paramètres passés au moment de la validation sont ajoutés à l'URL du script appelé.
   - Cette chaine s'écrit **`?nom1=valeur1&nom2=valeur2&...`**

Méthode GET
Donnez votre prénom, puis validez :
[ ] Valider
 	
Vous vous appelez
	`<FORM>`
	` Donnez votre prénom, puis validez :`
	`<INPUT TYPE=text NAME=`**`prenom`**`>`
	`<INPUT TYPE=submit VALUE="Valider">`
	`</FORM>`
	`<P>`
	` Vous vous appelez`
	`<?php`
	` echo "`**`$prenom`**`";`
	`?>`

Commentaires

   - L'interpréteur PHP a accès immédiat aux données des formulaires HTML, lorque ceux-ci sont soumis, c'est-à-dire à la validation par le bouton **`submit`**. La règle est simple :
     Le nom de l'élément de formulaire devient une variable PHP, qui porte le même nom, tout en étant préfixée par le symbole $ (ce qui caractérise toujours les variables PHP)

   - Si on laisse le champ de saisie vide, la page est rechargée, avec une URL à laquelle s'est ajouté **?nom=**.
     On en déduit que laisser le champ vide, n'a pas provoqué d'affectation à la variable PHP `$nom`, d'où l'affichage inchangé.
   - Si on a validé le nom "toto" dans le formulaire, la chaine **?nom=toto** est ajoutée à l'URL de la présente page , ce qui indique :
     1. Il y a bien affectation de `$nom` à la chaine saisie dans le champ nommé `nom`
     2. L'URL de la page étant changée, la page est rechargée
     3. L'instruction PHP incorporée dans le code HTML **`echo $nom;`** a été exécutée sur le serveur et se traduit par l'affichage observé
     4. Pour bien comprendre que la validation de la saisie équivaut à l'ajout de la chaine ?nom=.. dans l'URL, saisissez directement le nom dans l'URL.
        Qu'observez-vous ?
   - Le mode de requête décrit précédemment est le mode *par défaut*, appelé **`méthode GET`**. Les paramètres passés au moment de la validation sont ajoutés sous forme d'une chaine **`?nom1=valeur1&nom2=valeur2&...`** à l'URL demandée, qui se trouve être la même page par défaut, si on n'en précise pas d'autre.

2. **La méthode POST**

Méthode POST
Donnez votre prénom, puis validez :
[ ] Valider
 	
Vous vous appelez
	`<FORM ACTION="php-formulaire.php" METHOD="POST">`
	`Donnez votre prénom, puis validez :`
	`<INPUT TYPE=text NAME=prenom2>`
	`<INPUT TYPE=submit VALUE="Valider">`
	`</FORM>`
	`<P>`
	`Vous vous appelez`
	`<?php`
	`echo "$prenom2";`
	`?>`

Commentaires

Quelques fonctions et méthodes

- Il n'est pas possible de soumettre (par le bouton submit) conditionnellement le formulaire en fonction du contexte, par exemple suivant la valeur prise par une variable
  En effet, le fichier appelé, indiqué par la clause ACTION"..." de la balise FORM est unique. Ce script doit donc être capable de traiter tous les cas envisageables, fonction de l'état du formulaire à sa validation, et par conséquent doit pouvoir agir selon l'état de chaque composant du formulaire.
- On sait qu'il y a création automatique d'une variable PHP, pour chaque composant nommé (par NAME= ...), même si le composant n'a fait l'objet d'aucune saisie (champ "text"), ou d'aucune sélection (boutons, cases, listes déroulantes ..).
- Dans ce cas, ces variables de champ sont définies, mais n'ont pas de valeur. On teste cette situation avec la fonction **empty($var)**.
  qui alors renvoie TRUE. Sinon FALSE quand la variable $var est définie **et** possède une valeur.
- Pour savoir si une variable est définie, qu'elle ait reçue une valeur ou non, on utilise **isset($var)**.
  Elle renvoie TRUE si la variable $var est définie, FALSE sinon.
- Auto-soumission
  Il est plus simple finalement d'intégrer le script de traitement du formulaire dans la page même qui contient ce formulaire. Naturellement, cette page doit avoir l'extension **.php** pour que le serveur HTTP fasse appel à l'interpréteur PHP L'url passée dans le champ ACTION du formulaire doit donc être celle de la page elle-même.
- Affichage conditionnel d'un formulaire et de son traitement.
  On doit d'abord proposer le formulaire, puis lorsqu'il est soumis, ne pas le réafficher. Pour décider dans quel cas où se trouve il suffit de tester l'existence d'une variable dans le script, qui attestera à coup sûr que la validation du formulaire a bien eu lieu.
  Le plus simple est de tester la variable correspondant au nom du bouton **submit**
  Voici un schéma général de programme :

```
<?
if (isset($valide)) {
 /* traitement du formulaire */
} else {
?>
/ * affichage du formulaire HTML si $valide = FALSE */
<FORM ACTION="<? echo $PHP_SELF ?>"  METHOD="POST">
/* affichage des composants */
<INPUT TYPE="submit" NAME="valide" VALUE="Valider">
</FORM>
```

- Exemple de démonstration

Code PHP	Exécution
`<?` `if (isset($valide)) {` `if  (! empty($mot )` `  echo "Bonjour $mot <P>";` `else` `  echo "Vous n'avez pas donné votre prénom [:-( <P>" ;` `echo "Pour <A HREF=\"$PHP_SELF\"> afficher le formulaire</A>";` `}` `else` `echo "` `<FORM ACTION=\"$PHP_SELF\"  METHOD=\"POST\">` `Quel est votre prénom ?  ` `<input type=\"text\" name=\"mot\" value=\"\">` `<P>` `<INPUT TYPE=\"submit\" NAME=\"valide\" VALUE=\"Valider\">` `<INPUT TYPE=\"reset\" NAME=\"reset\" VALUE=\"Annuler\">` `</FORM>` `"` `?>`	Quel est votre prénom ? [ ] Valider Annuler

- Commentaires
  Remarquer comment on peut générer du code HTML à l'intérieur de PHP avec echo " .....", le code HTML pouvant se répartir sur plusieurs lignes.

Traitement des champs de formulaire

On suppose ici connues les balises des composants usuels d'un formulaire HTML. Si nécessaire, parcourir cette page d'initiation.

Les boutons radio

La variable PHP construite avec NAME reçoit la valeur associée au bouton sélectionné.

Code PHP

```
<form action="php-formulaire.php"  method="post">
<h4>Quel est votre niveau en programmation ?</h4>
<input type="radio" name="niveau" value="débutant">débutant
<input type="radio" name="niveau" value="moyen" checked>moyen
<input type="radio" name="niveau" value="expert">expert
<P>
<input type="submit" value="Valider">
</form>
<?php
echo "Etes vous sûr(e) d'avoir le niveau $niveau ?"
?>
```

Exécution

Quel est votre niveau en programmation ?

○débutant ◉moyen ○ expert

Valider

Etes vous sûr(e) d'avoir le niveau ?

Les cases à cocher

Il s'agit ici de composants indépendants, qui doivent donc porter des noms différents. Il est possible d'utiliser un nom générique de tableau pour ces cases, qui seront différenciés par un indice. Dans l'exemple ci-dessous les choix sont stockés dans un tableau appelé **choix[]**

Code PHP

```
<FORM action="php-formulaire.php" method="post">
Indiquez toutes vos couleurs préférées<br>
<TABLE><TR>
<TD bgcolor="red" width="50"> </TD><TD
<input type="checkbox" name="choix[]" value="rouge">Rouge
</TD></TR><TR>
<TD bgcolor="blue" width="50"> </TD><TD
<input type="checkbox" name="choix[]" value="bleu">Bleu
</TD></TR><TR>
<TD bgcolor="green" width="50"> </TD><TD
<input type="checkbox" name="choix[]" value="vert">Vert
</TD></TR><TR>
<TD bgcolor="black" width="50"> </TD><TD
<input type="checkbox" name="choix[]" value="noir">Noir
</TD></TR></TABLE>
<input type="submit" value="Valider">
</FORM>
<?
echo "<h4>Vos couleurs préférées sont </h4>";
for ($i=0 ; $i< sizeof($choix) ; $i++)
   if (isset($choix[$i]) )
   echo("$choix[$i]  - ");
?>
```

Exécution

Indiquez toutes vos couleurs préférées

☐Rouge
☐Bleu
☐Vert
☐Noir

Valider

Vous avez choisi :

Les composants image

Exercice de synthèse : inscription.php